

CLAIMS

1. A computer-implemented method, comprising:

running a debugging system on a local site, the local site being separated from a remote site by at least one firewall;

5 storing data on the local site, the data including a local copy of a remote application that is running on a virtual machine located on the remote site; and

making it appear to the debugging system that the remote application is running on the local site instead of the remote site by:

establishing a communication link between a first router located on the local site
10 and a second router located on the remote site;

using the communication link between the first and second routers to establish communication between the debugging system and the virtual machine;

receiving a timestamp indicating when the remote application was last modified;

15 using the received timestamp to determine whether the local copy is up to date with respect to the remote application; and

loading the local copy into the debugging system if the local copy is up to date, and otherwise (i) comparing the local copy with the remote application to establish delta information that identifies differences between the local copy and remote application, (ii) retrieving the delta information from the remote site, (iii) using the retrieved delta
20 information to alter the local copy to match the remote application, and (iv) loading the altered local copy into the debugging system.

2. The method of claim 1, wherein the remote application is a component of a larger application that is running on the virtual machine.

3. The method of claim 1, wherein the virtual machine is a Java virtual machine.

25 4. The method of claim 1, wherein the at least one firewall comprises a first firewall protecting the local site and a second firewall protecting the remote site.

5. The method of claim 1, further comprising:

sending commands from the debugging system to the virtual machine using the communication link.

6. The method of claim 1, further comprising:

receiving run-time data and state information about the remote application from the virtual machine through the communication link.

7. A computer program product, tangibly embodied in an information carrier, the computer program product being operable to cause data processing apparatus to:

run a debugging system on a local site, the local site being separated from a remote site by at least one firewall;

store data on the local site, the data including a local copy of a remote application that is running on a virtual machine located on the remote site; and

make it appear to the debugging system that the remote application is running on the local site instead of the remote site by:

establishing a communication link between a first router located on the local site and a second router located on the remote site;

using the communication link between the first and second routers to establish communication between the debugging system and the virtual machine;

receiving a timestamp indicating when the remote application was last modified;

using the received timestamp to determine whether the local copy is up to date with respect to the remote application; and

loading the local copy into the debugging system if the local copy is up to date, and otherwise (i) comparing the local copy with the remote application to establish delta information that identifies differences between the local copy and remote application, (ii) retrieving the delta information from the remote site, (iii) using the retrieved delta information to alter the local copy to match the remote application, and (iv) loading the altered local copy into the debugging system.

8. The product of claim 7, wherein the remote application is a component of a larger application that is running on the virtual machine.

9. The product of claim 7, wherein the virtual machine is a Java virtual machine.

10. The product of claim 7, wherein the at least one firewall comprises a first firewall protecting the local site and a second firewall protecting the remote site.

11. The product of claim 7, wherein the router is an SAProuter.

12. The product of claim 7, further comprising instructions to:

send commands from the debugging system to the virtual machine using the communication link.

5 13. The product of claim 7, further comprising instructions to:

receive run-time data and state information about the remote application from the virtual machine through the communication link.

14. A data processing system, comprising:

means for running a debugging system on a local site, the local site being separated from
10 a remote site by at least one firewall;

means for storing on the local site a local copy of a remote application that is running on a virtual machine located on the remote site; and

means for making it appear to the debugging system that the remote application is running on the local site instead of the remote site by:

15 establishing a communication link between a first router located on the local site and a second router located on the remote site;

using the communication link between the first and second routers to establish communication between the debugging system and the virtual machine;

receiving a timestamp indicating when the remote application was last modified;

20 using the received timestamp to determine whether the local copy is up to date with respect to the remote application; and

loading the local copy into the debugging system if the local copy is up to date, and otherwise (i) comparing the local copy with the remote application to establish delta information that identifies differences between the local copy and remote application,
25 (ii) retrieving the delta information from the remote site, (iii) using the retrieved delta information to alter the local copy to match the remote application, and (iv) loading the altered local copy into the debugging system.

15. The system of claim 14, wherein the remote application is a component of a larger application that is running on the virtual machine.

16. The system of claim 14, wherein the virtual machine is a Java virtual machine.

17. The system of claim 14, wherein the at least one firewall comprises a first firewall protecting the local site and a second firewall protecting the remote site.

18. The system of claim 14, wherein the router is an SAProuter.

5 19. The system of claim 14, further comprising:

means for sending commands from the debugging system to the virtual machine using the communication link.

20. The system of claim 14, further comprising:

10 means for receiving run-time data and state information about the remote application from the virtual machine through the communication link.